|  |  |  |
| --- | --- | --- |
| **Übung Nr. 2**  Jahrgang: BHME20  Gruppe: 3D |  | **Protokollabgabe**  Solldatum: 09.11.2023  Istdatum:  Note: |

PROTOKOLL

**Thema: Übung 7 & 8: Ansteuerung der 7-Segment Anzeige & RGB-LED**

Tag: 19.10.2023

Zeit: 10:45-13:15 Uhr

Ort: HTBLA Kaindorf | PRR Labor

Anwesend: Traußnigg Jan, Ursnik Iwana

Abwesend: -

Schriftführer\*in: Traußnigg Jan

Betreuer: Dipl.-Ing. Steiner Walter

**Aufgabenstellung**

Vervollständigen der Übung 7): Einlesen der gedrückten Buttons und Ausgabe bei den LEDs während die 7-Segment Anzeige weiterhin die Zahlen darstellt.

Übung 8) Ansteuerung einer RGB-LED: Alle Farben hintereinander eine Sekunde lang darstellen.

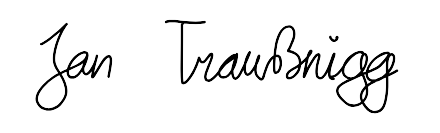
**Resümee**

In der heutigen Einheit durften wir lernen, wie wir zwei Prozesse gleichzeitig am Arduino ausführen können ohne dass uns das delay() einen Strich in die Rechnung macht. Zudem konnten wir lernen, wie man eine RGB-LED ansteuert und auch herausfinden wie man das herausfindet, wenn man es nicht weiß.

**Unterschriften**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Iwana Ursnik



\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Jan Traußnigg

[1 Zeitplan 2](#_Toc148370342)

[2 Thema 2](#_Toc148370343)

[2.1 Aufgabenstellung 2](#_Toc148370344)

[2.2 Verwendete Geräte und Hilfsmittel 2](#_Toc148370345)

[2.3 Vorgangsweise 2](#_Toc148370346)

[3 Messergebnisse 2](#_Toc148370347)

# Zeitplan

10:45 – 11:35 Wiederholung – Übung 7a)

11:35 – 12:25 Unterbrechung Labor

12:25 – 13:15 Wiederholung – Übung 7b)

# Thema

## Aufgabenstellung

**Übung 7**

1. Ändern Sie das Programm so ab, dass zusätzlich zum Zählen Übung 5 implementiert wird, also die Nummer der gedrückten Taste auf den Leds abgebildet wird.
2. Verbessern Sie das Programm so, dass die gedrückte Taste unmittelbar angezeigt wird und nicht erst mit einer Verzögerung von bis zu einer Sekunde.

Ein Bild, das Elektronik, Schaltung, Elektronisches Bauteil, Text enthält.

Automatisch generierte Beschreibung

Abbildung : Angabe Arduino Mega Shield   
(Pin-Nummern von dieser Abbildung ablesbar)

**Übung 8**

1. Schreiben Sie ein Programm, das die Farben der RGB-LED hintereinander im Sekundentakt darstellt. Die Farbnummer soll auf der 7 Segmentanzeige ausgegeben werden.
2. Erweitern Sie das Programm so, dass die Farbe nach dem Tastendruck dargestellt wird (KEY0 → Farbe 0, KEY1 → Farbe 1, … , KEY7 → Farbe
   1. Verwenden Sie die zuvor erstellten Funktionen (kbhit, getch und setSeg7).

Ein Bild, das Elektronik, Schaltung, Text, Elektronisches Bauteil enthält.

Automatisch generierte Beschreibung

Ein Bild, das Text, Screenshot, Diagramm, Schrift enthält.

Automatisch generierte Beschreibung

## Verwendete Geräte und Hilfsmittel

* Arduino IDE (privat-Laptop oder Schul-PC)
* Arduino UNO

## Vorgangsweise

### Übung 7) 3)

In der letzten Einheit wurde die Angabe ja so missverstanden, dass die Zahl an der 7 Segement-Anzeige geleuchtet hat, die bei den Tastern gedrückt wurde. Jetzt wird sie richtig gestellt, dass die 7 Segment Anzeige immer die Zahlen von 0-9 ausgibt und bei den LED’s die LED leuchtet, die man betätigt.

const int segmentPins[] = {62, 63, 64, 65, 66, 67, 68, 69};

const int buttonPins[] = {9, 8, 7, 6, 38, 39, 40, 41};

const int ledPins[] = {49, 48, 47, 46, 45, 44, 43, 42};

const int numbers[] = {

  B11000000,  // 0

  B11111001,  // 1

  B10100100,  // 2

  B10110000,  // 3

  B10011001,  // 4

  B10010010,  // 5

  B10000010,  // 6

  B11111000,  // 7

  B10000000,  // 8

  B10010000   // 9

};

void setSeg7(uint8\_t nr);

void setup() {

  for (int i = 0; i < 8; i++) {

    pinMode(segmentPins[i], OUTPUT);

    pinMode(ledPins[i], OUTPUT);

  }

  for (int i = 0; i < 8; i++) {

    pinMode(buttonPins[i], INPUT\_PULLUP);

  }

}

int buttonPressed;

void loop() {

  for (int k = 0; k <= 9; k++) {

    setSeg7(k);

    delay(1000);

  }

  buttonPressed = -1;

  for (int i = 0; i < 8; i++) {

    if (digitalRead(buttonPins[i]) == LOW) {

      buttonPressed = i;

      break;

    }

  }

  if (buttonPressed != -1) {

    for (int i = 0; i < 8; i++) {

      digitalWrite(ledPins[i], LOW);

    }

    digitalWrite(ledPins[buttonPressed], HIGH);

  }

}

void setSeg7(uint8\_t nr)

{

  for (int j = 0; j < 8; j++) {

    digitalWrite(segmentPins[j], bitRead(numbers[nr], j));

  }

}

### Übung 7) 4)

Erste Idee: Immer wenn (millis() % 1000) == 0 ist, nächste Ziffer ausgeben. Jedoch nach Test, wie lange ein Programmablauf braucht und dass dieser nicht immer bei 1000, 2000 etc. ankommt nochmal überdacht.

Ein Bild, das Text, Screenshot enthält.

Automatisch generierte Beschreibung

Ein Bild, das Text, Screenshot, Schrift, Zahl enthält.

Automatisch generierte Beschreibung

Zweite Idee: Immer wenn millis () den Meilenstein vom Intervall überschreitet nächste Ziffer ausgeben (der Rest des codes bleibt unverändert)

int buttonPressed;

int Interval = 1000;

int currentTime = Interval;

int currentNumber = 0;

void loop() {

  if(millis() >= currentTime) {

    currentTime += Interval;

    setSeg7(currentNumber++);

    if (currentNumber > 9) {

      currentNumber = 0;

    }

  }

  buttonPressed = -1;

  for (int i = 0; i < 8; i++) {

    if (digitalRead(buttonPins[i]) == LOW) {

      buttonPressed = i;

      break;

    }

  }

  if (buttonPressed != -1) {

    for (int i = 0; i < 8; i++) {

      digitalWrite(ledPins[i], LOW);

    }

    digitalWrite(ledPins[buttonPressed], HIGH);

  }

}

### Übung 8) 1)

Der erste Versuch war, die Licht-Ausgänge binär auf 0 oder 1 zu setzen doch nach rätseln, warum sich das Licht dabei kaum ändert, wurde die Erkenntnis gezogen, dass die RGB-LED beim Analogen Ausgang mit Werten von 0-255 bei jedem Eingang gesteuert wird. Dies wurde unter anderem auch durch die Anleitung von Wokwi und einem example-Projekt herausgefunden.

Ein Bild, das Text, Screenshot, Software, Display enthält.

Automatisch generierte Beschreibung

Ein Bild, das Text, Screenshot, Software, Computersymbol enthält.

Automatisch generierte Beschreibung

https://wokwi.com/projects/306455554559050306

Jedoch war hat auch mit dieser Erkenntnis noch ein Teil des nötigen Wissens gefehlt. Bei der Ansteuerung des roten Pins mit 255 leuchtet die Farbe Cyan, was mich zuerst sehr verwirrt hat. Jedoch bin ich nach ein bisschen ausprobieren und testen auf den Entschluss gekommen, dass die Pins bei 0 leuchten und bei 255 ausgeschaltet werden.

Ein Bild, das Text, Screenshot, Software, Computersymbol enthält.

Automatisch generierte Beschreibung

Somit kann nur der Code richtig gestellt werden:

int redPin = 5;

int greenPin = 3;

int bluePin = 2;

int colors[7][3] = {

    {0, 255, 255}, // Rot

    {0, 0, 255}, // Gelb

    {0, 0, 0}, // Weiß

    {255, 0, 255}, // Grün

    {255, 0, 0}, // Cyan

    {0, 255, 0}, // Magenta

    {255, 255, 0} // Blau

  };

void setup() {

  pinMode(redPin, OUTPUT);

  pinMode(greenPin, OUTPUT);

  pinMode(bluePin, OUTPUT);

}

void loop() {

  for (int i = 0; i < 7; i++) {

    analogWrite(redPin, colors[i][0]);

    analogWrite(greenPin, colors[i][1]);

    analogWrite(bluePin, colors[i][2]);

    delay(1000);

  }

}

### Übung 8) 2)

# Messergebnisse

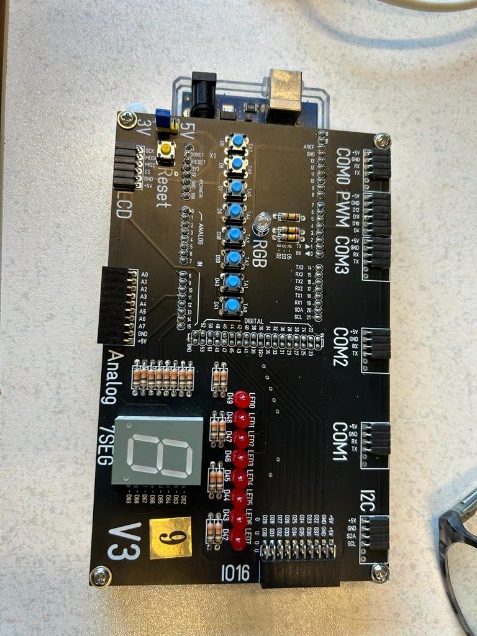


Abbildung : Arduino Mega Shield   
(Foto von Kollegin Anna-Aurora Schreiner)